CERTIFICATE OF TRANSMISSION OR MAILING

I hereby certify that this correspondence is being transmitted by either fax to the U.S. Patent and Trademark office to fax number 571-273-8300, or is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 to on April 11, 2008.

/Brian C. Kunzler/
Attorney for Applicant

PATENT
Docket No. 1500.2.14
Client Docket No. TUC920030050US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Yu-Cheng Hsu et al.	
Serial No.:	10/686,878	
Filed:	October 16, 2003	Group Art Unit: 2136
For:	APPARATUS SYSTEM AND METHOD FOR DETERMINISTICALLY TRANSFERRING DATA BY REBOOTING TO A DATA TRANSFER KERNEL	
Examiner:	Carlton Johnson	

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Examiner:

The USPTO received Appellant's Notice of Appeal on 27 December 2007, which was filed in response to an Advisory Action mailed 27 November 2007. Appellant appeals the rejection of pending claims 1, 3-8, 10-22, and 24-30.

This Brief is being filed under the provisions of 37 C.F.R. § 41.37. The filing fee set forth in 37 C.F.R. § 41.20(b)(2) of $510.00 was previously submitted. The Commissioner is hereby authorized to charge payment of any additional fees associated with this communication, or to credit any overpayment, to Deposit Account No. 090449.

1. REAL PARTY IN INTEREST

The real party in interest is the assignee, International Business Machines Corporation, Armonk, New York.

2. RELATED APPEALS AND INTERFERENCES

There are no related appeals, interferences, or judicial proceedings.

3. STATUS OF CLAIMS

The Office Action mailed on 07 August 2007 and the Advisory Action mailed 27 November 2007 maintained the rejection of claims 1, 10, 13, 17, 24, and 28 under 35 U.S.C. 112 for being indefinite with regard to the word "reboot" in the specification and claims, and the rejection of claims 1-7, 10, 11, 13-22, and 24-30 under 35 U.S.C. 102(e) as being anticipated by Backman et al. U.S. Patent No. 7,124,322 (hereinafter "Backman"), and the rejection of claims 8, 9, 12, 22, and 23 under 35 U.S.C. 103(a) as being unpatentable over Backman in view of Neuman et al. U.S. PGPUB No. 20030217299. Appellants appeal the rejection of all the claims but will focus arguments on independent claims 1, 10, 13, 17, 24, and 28.

4. STATUS OF AMENDMENTS

Appellants submitted a minor amendment to the claims on 05 November 2007 which was entered by the examiner on 27 November 2007.

5. SUMMARY OF CLAIMED SUBJECT MATTER

The claimed subject matter deals with a specific apparatus, system, method, and computer readable medium to rapidly and deterministically save data within volatile memory to a data storage device in response to an abnormal operating condition that threatens a loss of data in the volatile

memory. See the abstract of Appellant's Published Application No. US 2005/0086461 (hereinafter "Specification").

The following quotation of independent claims 1, 10, 13, 17, 24, and 28 include reference numerals and parenthetical references to representative examples of the elements and components recited in each claim in compliance with 37 CFR 41.37(c)(1)(v).

1. An apparatus for rapidly, deterministically transferring data, the apparatus comprising:

a processor (310) configured to process data; [*see, for example, paragraph 44*]

a volatile memory (320) configured to store the data; [*see, for example, paragraph 44*]

a boot control module (330) configured to boot the processor with a standard operating kernel (520) under a normal operating condition and to reboot the processor with a data transfer kernel (620) under an abnormal operating condition that threatens a loss of data in the volatile memory; and [*see, for example, paragraphs 45-48, 49-51, and 55-56*]

the data transfer kernel (620) configured to support a data save operation configured to save data in the volatile memory to a storage device. [*see, for example, paragraphs 46-47 and 55-56*]

10. An apparatus for rapidly, deterministically transferring data to a storage device, the apparatus comprising:

a storage device (350) configured to store data; [*see, for example, paragraph 47*]

a data transfer kernel (620) configured to support data saving operations; and [*see, for example, paragraphs 46-47 and 55-56*]

a computer (upper box of Figure 3) in communication with the storage device, the computer configured to load the data transfer kernel during a reboot procedure in response to an abnormal operating condition that threatens the loss of data in a volatile memory; the data transfer kernel configured to support a data save operation configured to save data in the volatile memory to the storage device. [*see, for example, paragraphs 45-48, 49-51, and 55-56*]

13. An apparatus for rapidly, deterministically saving data, the apparatus comprising:

means for saving data (340) in a non-volatile memory; [*see, for example, paragraph 46*]

means for detecting a data save condition (410) comprising an abnormal operating condition that threatens the loss of data in a volatile memory; and [*see, for example, paragraph 50*]

means for booting a processor (330) with a data transfer kernel in response to the abnormal operating condition, the data transfer kernel configured to save data to the means for saving data. [*see, for example, paragraphs 45-48, 49-51, and 55-56*]

17. A system for rapidly, deterministically saving data to a storage device, the system comprising:

a processor (310) configured to process data; [*see, for example, paragraph 44*]

a memory (320) configured to provide volatile storage for the data; [*see, for example, paragraph 44*]

a storage device (350) configured to provide non-volatile storage for the data; and [*see, for example, paragraph 47*]

a boot control module (330) configured to boot the processor module with a standard operating kernel under a normal operating condition and to reboot the processor with a data transfer kernel under an abnormal operating condition that threatens the loss of data in the memory; the data transfer kernel configured to support a data save operation configured to save data in the memory to the storage device. [*see, for example, paragraphs 45-48, 49-51, and 55-56*]

24. A method for rapidly, deterministically saving data, the method comprising:

detecting a data save condition (Figure 2 step 120) that threatens the loss of data in a volatile memory; and [*see, for example, paragraph 38*]

rebooting a processor module (Figure 2 step 210) with a data transfer kernel (620) configured to support a data save operation configured to save the data in the volatile memory to a non-volatile storage device. [*see, for example, paragraphs 39, 45-48, 49-51, and 55-56*]

28. A computer readable storage medium comprising computer readable program code for rapidly, deterministically saving data, the program code configured to:

boot a processor module (Figure 2 step 210) with a data transfer kernel (620) configured to support a data save operation (Figure 2 step 240) and in response to an abnormal operating condition that threatens the loss of data in a volatile memory module comprising volatile memory; and [*see, for example, paragraphs 38 and 39*]

transfer the data with the data save operation from the ~~a~~ memory module (320) to a non-volatile storage device (350). [*see, for example, paragraphs 39, 45-48, 49-51, and 55-56*]

The present invention in the various embodiments presented in the foregoing claims, enables the rapid and deterministic saving of data from a volatile memory to a (non-volatile) storage device in response to detecting an abnormal operating condition that threatens a loss of data in the volatile memory. The rapid and deterministic saving of data is accomplished by rebooting a processor with a data transfer kernel that supports data save operations. Rebooting clears the processor of executing processes and frees the processor to execute the data save operation(s) supported by the data transfer kernel. A data transfer kernel is distinguished from a standard operating kernel used in normal operation in that a data transfer kernel supports a limited set of software and hardware processes that are required to save data such as configuring a processor and storage devices for data save (*i.e* data transfer) operations and loading and executing a data transfer process to conduct the data save operations [see paragraphs 39, 55, and 56].

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Whether the Examiner has properly rejected claims 1, 10, 13, 17, 24, and 28 under 35 U.S.C. 112 for being indefinite with regard to the word "reboot" in the specification.

II. Whether the Examiner has established a *prima facie* case of anticipation of claims 1, 10, 13, 17, 24, and 28 under 35 U.S.C. 102(e) by Backman et al. U.S. Patent No. 7,124,322 (hereinafter "Backman").

7. ARGUMENT

I. The Examiner has improperly rejected claims 1, 10, 13, 17, 24, and 28 under 35 U.S.C. 112 for being indefinite with regard to the word "reboot" in the specification and claims.

Appellants submit that claim 1 is representative of the subject matter recited in independent claims 10, 13, 17, 24 and 28. Therefore, Appellant's response will focus on claim 1 with the understanding that responses for claims 10, 13, 17, 24, and 28 would follow a similar vein.

Claim 1 recites:

> 1. (Previously Amended) An apparatus for rapidly, deterministically transferring data, the apparatus comprising:
>
> a processor configured to process data;
>
> a volatile memory configured to store the data; and
>
> a boot control module configured to boot the processor with a standard operating kernel under a normal operating condition and to reboot the processor with a data transfer kernel under an abnormal operating condition that threatens a loss of data in the volatile memory;
>
> the data transfer kernel configured to support a data save operation configured to save data in the volatile memory to a storage device.

The Examiner maintains that the common definition of reboot includes interpretations that would render the invention inoperable and for that reason the claims and specification are indefinite. At issue is whether the term reboot refers to a warm boot procedure in which booting occurs without loss of power or a cold boot procedure in which power to the apparatus is cycled and whether a cold boot procedure would be inoperable since the working memory may be erased during a cold boot procedure.

In response to the Examiner's rejection, Appellants assert the following:

- The examiners evidence provides equal support for cold boot and warm boot interpretations of the term "reboot".
- It is not necessary or prudent to eliminate all inoperable embodiments from the scope of valid claims.
- One of skill in the art would consider a warm boot procedure as the preferred embodiment of a reboot procedure.
- A cold boot procedure is operable in certain embodiments.
- Selection of a cold boot or a warm boot procedure is best left to one of skill in the art.

The following paragraphs explore these assertions in greater detail.

The Examiner indicated that 5 of the 6 definitions of reboot found by the Examiner include a cold boot within the scope of the term reboot while only 1 of the 6 definitions exclusively included only a warm boot interpretation of reboot. Appellants assert that the assertion is misleading and note that 5 of the 6 definitions of reboot provided by the Examiner support the warm boot interpretation and 1 of the 6 sources expressly excluded a cold boot interpretation. Appellants assert that a scoring of the definitions provided by the Examiner would be 5 out of 6 for both a cold boot interpretation and a warm boot interpretation. Appellants therefore assert that the Examiner has provided evidence that a warm boot is the valid interpretation of the term 'reboot'. Appellants therefore assert that the Examiner has provided evidence that operable versions of the invention fit within the scope of the claims as interpreted by both the Examiner and the Appellants.

Appellants question whether it is necessary (or legal to require) that the language of a claim explicitly exclude inoperable interpretations or embodiments. Appellants submit that virtually any invention could be embodied in an inoperable form and still fit within the scope of the claims of an issued patent. Appellants ask if it would be proper to invalidate each issued patent that includes an

inoperable embodiment within the scope of the claims of the issued patent. Appellants note that there is no need to protect the public from inoperable embodiments by denying an Applicant intellectual property that includes inoperable embodiments since inoperable embodiments are of no value to society and no consequence in the marketplace.

Returning to the issue at hand, Appellants assert that the Examiner has erroneously assumed an orientation of one that is not skilled in the art in dealing with the memory retention issues related to a reboot. It is self evident that one of skill in the art would not desire to create an inoperable embodiment and would select or design an embodiment that would render the invention operable. Furthermore, the language of claim 1 expressly includes *"a boot control module configured to ... reboot the processor with a data transfer kernel under an abnormal operating condition that threatens a loss of data in the volatile memory."* Appellants assert that it is apparent from the claims language that the reboot procedure should not destroy the data in the volatile memory, particularly since the culmination and focus of the claim is "*to save data in the volatile memory to a storage device*". Therefore, one of skill in the art under guidance of the teachings of the specification and the context of the claims would ensure that the claimed invention would preserve the data within the volatile memory during a reboot procedure. Appellants therefore assert that the Examiner's concerns are unfounded.

Turning to the Examiners interpretation of reboot, Appellants note that while it may be conventional for humans to cycle power to reboot a computer, it is not conventional (though it may be possible in certain embodiments) for a "boot control module" or any other module to cycle power to reboot a computer. As defined in paragraphs 28-30 of Appellant's specification, modules may be embodied as hardware, software, or some combination thereof. Appellants assert that conventional embodiments of such modules cannot perform their intended function in an un-powered state. Therefore, one of skill in the art would typically assume that an apparatus generally and the boot control module specifically would only operate in a powered state. Furthermore, since most power

supplies are not electronically controllable (probably due to the cost of providing electronic control), one of skill in the art would generally assume that power would not be cycled during a reboot procedure initiated by a boot control module particularly if one desires to protect data within volatile memory.

Appellants note that the claims specify that the reboot control module is configured to reboot the processor and that most processors have a reset control signal and/or a reset instruction that resets the program counter to a known location in order to commence a boot procedure and reboot the processor and the computer. Furthermore, the ability to electronically control power to an apparatus typically requires adding expensive power transistors or electronically controlled switches that increase the overall cost of a system. Such added cost is unnecessary since processors are readily able to conduct a reboot procedure without cycling power. Appellants submit that these issues are common knowledge to those of skill in the art and would provide context that would virtually eliminate the inoperable interpretation of reboot put forth by the Examiner.

Regarding the assertion that a cold boot procedure is inherently inoperable, Appellants request that the Board consider whether all embodiments of a cold boot procedure would erase working (i.e. volatile) memory and render the invention inoperable. Appellants note that is well known that working memory is not immediately erased in many computing systems in response to turning off the power to the computer. This is due to a variety of factors such as the capacitance of the power supply, the inclusion of bypass capacitors next to each chip on each printed circuit board in order to reduce power supply noise, and the inherent capacitance of memory cells. As a result, data may be retained in memory in many electronic devices for many seconds beyond the removal of power.

Early users of personal computers were very familiar with the need to turn off a computer to clear the working memory of the computer in response to a system crash or a virus. Even today, Symantec corporation has the following instructions on their web site (see

http://www.symantec.com/security_response/writeup.jsp?docid=2003-081915-0030-99) when dealing with a certain virus:

> Note: If, when running the tool, you see a message that the tool was not able to remove one or more files, run the tool in Safe mode. Shut down the computer, turn off the power, and wait 30 seconds. Restart the computer in Safe mode and run the tool again. All the Windows 32-bit operating systems, except Windows NT, can be restarted in Safe mode. For instructions on restarting the computer in Safe mode, read the document, How to start the computer in Safe Mode.

Note the necessity of turning off the computer for an extended length of time to ensure that the working (*i.e.* volatile) memory is cleared and the virus is no longer present within the working memory of the computer. Appellants assert that the assumption that a cold boot procedure would be inoperable is not true in some cases. Appellants acknowledge that rebooting with a cold boot procedure is probably not preferred but also admit that it may be preferred in certain circumstances. Appellants assert that selection of a cold boot or a warm boot procedure and the details associated therewith is within the working knowledge of one of skill in the art and that the specification need not disclose such details to be enabling.

In summary, Appellants assert that the examiners evidence provides equal support for cold boot and warm boot interpretations of a reboot procedure, it is not necessary or prudent to eliminate inoperable embodiments from the scope of valid claims, that one of skill in the art would consider a warm boot procedure as the preferred embodiment of a reboot procedure, that a cold boot procedure could be operable, and that selection of a cold boot or a warm boot procedure is within the working knowledge of one of skill in the art. Therefore Appellants request that the Board acknowledge that the current language of the claims and specification is sufficiently definite and enabling, and that the rejection under 35 USC 112 is improper.

II. The Examiner has not established a *prima facie* case of anticipation of claims 1, 10, 13, 17, 24, and 28 under 35 U.S.C. 102(e) by Backman et al. U.S. Patent No. 7,124,322.

A review of the present invention may help clarify the novelty of Appellants' claims over Backman. Referring to the written descriptions of Figures 1 and 2 of the application, a processor is booted with a standard operating kernel that supports normal operating conditions. Under normal operating conditions, data is stored in a volatile memory. If an abnormal operating condition that threatens the loss of data in the volatile memory is detected, the processor is rebooted with a data transfer kernel. Rebooting the processor clears the processor of any previously running processes and thereby enables the processor to run a data save operation supported by the data transfer kernel. The data save operation saves the data located in the volatile memory to a non-volatile storage device.

Appellants arguments will focus on the elements of independent claim 1 with the understanding that the same arguments apply to the elements delineated in independent claims 10, 13, 17, 24, and 28.

Regarding the rejection of claim 1 under 35 USC 102(e), Appellants assert that Backman does not disclose the claimed limitation of "data transfer kernel configured to support a data save operation". Appellants note the follow definition of Kernel from Webopedia:

> Kernel
>
> The central module of an operating system. It is the part of the operating system that loads first, and it remains in main memory. Because it stays in memory, it is important for the kernel to be as small as possible while still providing all the essential services required by other parts of the operating system and applications. Typically, the kernel is responsible for memory management, process and task management, and disk management.

While Backman does disclose a data backup utility or application, Appellants assert that Backman does not disclose "a data transfer kernel configured to support a data save operation". The

use of a kernel dedicated to data save operations provides additional functionality over a utility or application including immediate accessibility, a deterministic execution time, and faster execution since the computer does not need to continually switch context between the operating system kernel and an application or utility. For example, a typical computer must switch state between the supervisor privileges associated with ring 0 (*i.e.* kernel) processes and the user privileges of outer ring processes that are executed by applications and utilities. The overhead for such context switching can considerably reduce performance and make the computing system more vulnerable to malware such as viruses.

Appellants assert that the cited prior art does not suggest or teach a motivation for providing a data transfer kernel configured to support a data save operation. Appellants therefore assert that Backman discloses a solution for saving data distinct in operation, objective, and perspective from the solution disclosed by the present invention. Referring to column 5 lines 56-67 and column 6 lines 25-45, Backman teaches a data restoration solution that includes backing up data stored on a work station hard drive through an imaging process during normal operations, rebooting the workstation to a restoring environment in response to a massive data loss or failure of the hard drive, and restoring the hard drive from the data images produced by the aforementioned imaging process. Therefore, Appellants' read Backman as disclosing a hard drive restoration solution via previously generated data images.

In contrast to Backman, Appellants disclose a solution for saving data in volatile memory to a non-volatile memory in response to abnormal operating conditions via a data transfer kernel. As such, Appellants respectfully assert that Backman and the present invention cover processes that are not only distinct in terms of claim elements, but are distinct in perspective and objectives as well.

In addition to not disclosing a data transfer kernel, Appellants submit that Backman does not disclose "a boot control module configured to ... reboot the processor with a data transfer kernel under an abnormal operating condition that threatens a loss of data in the volatile memory". Appellants submit that the rebooting that occurs in Backman is conducted by an administrator or user in order to restore previously saved data images see column 6 lines 25-44 and is not conducted automatically by a boot control module "under an abnormal operating condition that threatens a loss of data in the volatile memory".

In addition to not disclosing a boot control module and a data transfer kernel, Appellants submit that Backman does not teach the interactions specified in Appellants claims. For example, claim 24 reads "detecting a data save condition that threatens the loss of data in a volatile memory; and rebooting a processor module with a data transfer kernel configured to support a data save operation configured to save the data in the volatile memory to a non-volatile storage device." This interaction is intrinsic to the elements included in claim 1 which reads "a boot control module configured ... to reboot the processor with a data transfer kernel under an abnormal operating condition that threatens a loss of data in the volatile memory; and the data transfer kernel configured to support a data save operation configured to save data in the volatile memory to a storage device."

The interaction, sequencing, and functional elements taught by Backman are completely different in objective, intent, and substance than the interaction, sequencing, and functional elements specified or implied by Appellants' claims and taught in Appellants' specification. Specifically, Backman teaches saving hard drive image data on a predetermined basis during normal operations and manually restoring data during normal operations after a disaster has occurred that destroys the data on hard drives associated with the computers of one or more users (see for example Figure 3, column 4 lines 38-39, and column 6 lines 25-29). Appellants submit that Backman teaches away from the present invention in that saving data in Backman does not occur under a condition that threatens the loss of data in volatile memory. Appellants also assert that the data saving that occurs in Backman relates to hard drive images and not to data in volatile memory. Appellants also assert that rebooting in Backman occurs manually and is related to restoring data to non-volatile storage devices rather that saving data in volatile memory that may be lost if it is not quickly saved (see column 6 lines 25-29).

Given the foregoing, Appellants respectfully assert that Backman fails to anticipate the present invention and also fails to provide a motivation to combine other references to render the present claims obvious. More particularly, Backman fails to disclose, suggest, teach, or provide a motivation for rapidly, deterministically transferring data that includes rebooting a processor with a data transfer kernel under an abnormal operating condition that threatens a loss of data in volatile memory and saving the data located in the volatile memory to a storage device via a data save operation.

SUMMARY

In view of the foregoing, Appellants respectfully assert that each of the claims on appeal have been improperly rejected because the Examiner has failed to show that Backman anticipates each of the limitations of independent claims 1, 10, 13, 17, 24, and 28. Furthermore, Appellants assert that the Examiners assertion that the specification was not enabling was based on faulty assumptions.

Respectfully submitted,

Date: April 11, 2008

/Brian C. Kunzler/

Kunzler & McKenzie
8 E. Broadway, Suite 600
Salt Lake City, Utah 84111
Telephone: 801/994-4646

Brian C. Kunzler
Reg. No. 38,527
Attorney for Applicant

8. CLAIMS APPENDIX

Claims involved in the appeal

1. (Previously Amended) An apparatus for rapidly, deterministically transferring data, the apparatus comprising:

a processor configured to process data;

a volatile memory configured to store the data;

a boot control module configured to boot the processor with a standard operating kernel under a normal operating condition and to reboot the processor with a data transfer kernel under an abnormal operating condition that threatens a loss of data in the volatile memory; and

the data transfer kernel configured to support a data save operation configured to save data in the volatile memory to a storage device.

2. (Canceled)

3. (Previously Amended) The apparatus of claim 1, wherein the data save operation is selected from the group consisting of a storage configuration operation, a transfer process loading operation, a data transfer operation, and a system shutdown operation.

4. (Original) The apparatus of claim 3, wherein the data transfer kernel is configured to exclusively support the data save operation.

5. (Previously Amended) The apparatus of claim 1, further comprising a memory module comprising data bits for marking data to be saved during the data save operation.

6. (Original) The apparatus of claim 5, wherein the standard operating kernel is further configured to mark data to be saved during a data save operation.

7. (Original) The apparatus of claim 1, wherein the data transfer kernel is configured to configure the storage device for specialized data save operations.

8. (Original) The apparatus of claim 1, wherein the data transfer kernel is configured to conduct a power down procedure.

9. (Canceled)

10. (Previously Amended) An apparatus for rapidly, deterministically transferring data to a storage device, the apparatus comprising:

a storage device configured to store data;

a data transfer kernel configured to support data saving operations; and

a computer in communication with the storage device, the computer configured to load the data transfer kernel during a reboot procedure in response to an abnormal operating condition that threatens the loss of data in a volatile memory; the data transfer kernel configured to support a data save operation configured to save data in the volatile memory to the storage device.

11. (Original) The apparatus of claim 10, wherein the data transfer kernel is configured to exclusively support devices and processes required to save data to the storage device.

12. (Original) The apparatus of claim 10, wherein the data transfer kernel is configured to power down the computer and the storage device.

13. (Previously Amended) An apparatus for rapidly, deterministically saving data, the apparatus comprising:

means for saving data in a non-volatile memory;

means for detecting a data save condition comprising an abnormal operating condition that threatens the loss of data in a volatile memory; and

means for booting a processor with a data transfer kernel in response to the abnormal operating condition, the data transfer kernel configured to save data to the means for saving data.

14. (Original) The apparatus of claim 13, further comprising means for configuring the means for saving data for data save operations.

15. (Original) The apparatus of claim 13, further comprising means for booting a standard operating kernel for normal operation.

16. (Original) The apparatus of claim 13, further comprising means for marking data to be saved during a data save operation.

17. (Previously Amended) A system for rapidly, deterministically saving data to a storage device, the system comprising:

a processor configured to process data;

a memory configured to provide volatile storage for the data;

a storage device configured to provide non-volatile storage for the data; and

a boot control module configured to boot the processor module with a standard operating kernel under a normal operating condition and to reboot the processor with a data transfer kernel under an abnormal operating condition that threatens the loss of data in the memory; the data

transfer kernel configured to support a data save operation configured to save data in the memory to the storage device.

18. (Previously Amended) The system of claim 17, wherein the standard operating kernel is configured to mark data in the memory to be saved by the data transfer kernel during a data save operation.

19. (Previously Amended) The system of claim 17, wherein the data transfer kernel exclusively supports devices, operations, and processes required to save data.

20. (Original) The system of claim 17, wherein the data transfer kernel configures the processor for data saving operations.

21. (Original) The system of claim 17, wherein the data transfer kernel configures the storage device for specialized data saving operations,

22. (Original) The system of claim 17, wherein the data transfer kernel is configured to conduct a power down procedure.

23. (Canceled)

24. (Previously Amended) A method for rapidly, deterministically saving data, the method comprising:

detecting a data save condition that threatens the loss of data in a volatile memory; and

rebooting a processor module with a data transfer kernel configured to support a data save operation configured to save the data in the volatile memory to a non-volatile storage device.

25. (Previously Amended) The method of claim 24, further comprising exclusively supporting devices, operations, and conducting processes required to save data to a storage device.

26. (Previously Amended) The method of claim 24, further comprising configuring the non-volatile storage device to receive data.

27. (Original) The method of claim 24, further comprising marking data to be saved by the data transfer kernel.

28. (Previously Amended) A computer readable storage medium comprising computer readable program code for rapidly, deterministically saving data, the program code configured to:

boot a processor module with a data transfer kernel configured to support a data save operation and in response to an abnormal operating condition that threatens the loss of data in a volatile memory module comprising volatile memory; and

transfer the data with the data save opeartion from the ~~a~~ memory module to a non-volatile storage device.

29. (Original) The computer readable storage medium of claim 28, wherein the computer readable code is further configured to mark data in the memory module to be saved to the storage device.

30. (Previously Amended) The computer readable storage medium of claim 28, wherein the computer readable code is further configured to exclusively support devices, operations, and processes required to save data to the storage device.

9. EVIDENCE APPENDIX

The following definitions of reboot were entered into the file with the response of 07 August 2007:

1. http://www.scala.com/definition/reboot.html which defines reboot as:

   "To reboot your computer is to restart it without turning off the power. This is also known as a "warm boot"."

2. http://www.webopedia.com/TERM/R/reboot.html which defines reboot as:

   "To restart a computer. In DOS, you can reboot by pressing the Alt, Control and Delete keys simultaneously. This is called a warm boot. You can also perform a cold boot by turning the computer off and then on again."

3. http://www.allwords.com/word-reboot.html which defines reboot as:

   (*computing*) To cause a computer to execute its boot process, effectively resetting the computer and causing the operating system to reload, especially after a system or power failure."

4. http://whatis.techtarget.com/ which defines reboot as:

   "To reboot is to restart a computer and reload the operating system. The most common reasons to reboot are because the installation of new software or hardware requires it, or because applications are not responding for some reason."

5. http://www.yourdictionary.com/reboot which defines reboot as:

   "to boot again, as to restore the computer to operation after a program failure"

6. http://www.thefreedictionary.com/reboot which defines reboot as:

   "To turn (a computer or operating system) off and then on again; restart."

10. RELATED PROCEEDINGS APPENDIX

None.